อาจหาญ สัตยารักษ์

SOFT MAIL

การทำข้อสอบ ปรนัย 4 ตัวเลือก ตอนที่ 1 สร้างเครื่องคิดเลข

เป็นเทคนิควิธีการในการเขียนโปรแกรมสร้างเครื่องคิดเลขเพื่อใช้ในการทำข้อสอบ

โปรแกรมทำข้อสอบส่วนมากนั้นผู้สอบมักจะไม่สามารถทำข้อสอบให้เหมือนจริงได้ เช่น ไม่สามารถข้ามทำข้อสอบข้อใดข้อหนึ่งไปก่อน หรือไม่สามารถแก้ไขการตอบข้อที่ตอบไปก่อนหน้านี้ได้ ฯลฯ แม้แต่โปรแกรมที่ผู้เขียนพัฒนาขึ้นมา เช่น โปรแกรมกราฟิก THAISHOW โปรแกรม "ไทยทัศน์" (โปรแกรมนี้เป็นงานวิจัยที่ได้รับเงินสนับสนุนของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)) โปรแกรมข้อสอบปรนัยก็ต้องทำเรียงลำดับไปครั้งละข้อเช่นกัน

ผู้เขียนได้รับคำร้องขอจากผู้อ่านหลายท่านให้เขียนโปรแกรมข้อสอบปรนัยที่ทำงานได้เหมือนจริง ซึ่งผู้เขียนก็ตอบสนองโดยขอเสนอบทความชุดนี้ในรูปของโครงงาน โดยบทความชุดนี้มีทั้งหมด 3 ตอน คือ

ตอนที่ 1 สร้างเครื่องคิดเลข
ตอนที่ 2 สร้างแบบทดสอบ
ตอนที่ 3 การสอบด้วยแบบทดสอบ

ผู้อ่านคงต้องติดตามบทความนี้ไปจนครบ 3 ตอน จึงจะได้โปรแกรมทำข้อสอบที่เหมือนจริงไปใช้งาน

ซอฟต์เมล์ของตอนที่ 1 นี้ประกอบด้วยไฟล์ 6 ไฟล์ คือ thai.tpu, head.pas, calculat.pas, litt.obj, minical.pas และ minical.exe

ไฟล์ thai.tpu นั้นมาจากวารสารไมโครคอมพิวเตอร์ฉบับที่ 104 รูปที่ 5 หน้า 252 จากบทความเรื่อง เทคนิคการนำไฟล์ข้อมูลไปไว้ในไฟล์ EXE

ไฟล์ head.pas ต้นฉบับมาจากบทความเรื่อง เมนู 3 มิติกับเม้าส์ในวารสารไมโครคอมพิวเตอร์ฉบับที่ 105 รูปที่ 1 และ 2 โดยผู้เขียนขอปรับปรุงใหม่ดังที่จะกล่าวถึงต่อไป

ดังนั้นผู้อ่านจะต้องย้อนไปดูรายละเอียดในวารสารทั้งสองฉบับอีกครั้งหนึ่งด้วย

ปรับปรุงไฟล์เกี่ยวกับเม้าส์

เนื่องจากยูนิต head.tpu ในวารสารไมโครคอมพิวเตอร์ฉบับที่ 105 เรื่อง "เมนู 3 มิติกับเม้าส์" มีปัญหาบางประการคือ ถ้ามีการกดปุ่มของเม้าส์แช่ไว้จะมีการทำงานซ้ำ ซึ่งแม้จะใช้การหน่วงเวลาด้วยคำสั่ง delay แล้วก็อาจมีปัญหาในการทำงานได้

ผู้เขียนจึงขอปรับปรุงไฟล์ head.pas (ซึ่งคือยูนิต head.tpu) ให้สามารถแก้ปัญหานี้ได้ ซึ่งต้องปรับปรุงดังนี้ ตัดบรรทัดที่ 81-87 ในรูปที่ 2 ออกไป แล้วนำโปรแกรมในรูปที่ 1 ฉบับนี้ไปแทนที่ พร้อมกับแทรกหลังบรรทัดที่ 12 ในรูปที่ 1 อีกหนึ่งบรรทัด คือ PROCEDURE ClearMouse; เพื่อให้ชุดคำสั่งนี้สามารถเรียกใช้จากภายนอกยูนิต head.tpu ได้

ส่วนที่ปรับปรุงเป็นดังรูปที่ 1 มีรายละเอียดในการปรับปรุงไฟล์ head.pas ของเดิมดังนี้

1. บรรทัดที่ 1 เพิ่มตัวแปร 4 ตัวแปรเพื่อรองรับขอบเขตหน้าต่างของเม้าส์ที่กำหนดไว้

2. บรรทัดที่ 2-10 ปรับปรุงชุดคำสั่ง WindowMouse ใหม่ โดยเพิ่มส่วนบรรทัดที่ 8-9 เพื่อจำขอบเขตของหน้าต่างเม้าส์ที่กำหนดไว้ในตัวแปรใหม่ในบรรทัดที่ 1

3. บรรทัดที่ 11-19 ชุดคำสั่งที่เพิ่มเพื่อทำการยกเลิกการกดปุ่มเม้าส์ไม่ให้มีการทำงานเมื่อกดปุ่มของเม้าส์แช่ไว้

4. บรรทัดที่ 14 หาตำแหน่งเคอร์เซอร์อยู่มาเก็บไว้ในตัวแปร x,y

5. บรรทัดที่ 15 เคลียร์การกดปุ่มของเม้าส์

6. บรรทัดที่ 16 กำหนดขอบเขตหน้าต่างของเม้าส์ให้อยู่ในขอบเขตเดิมที่กำหนดไว้แล้ว

7. บรรทัดที่ 17 กำหนดตำแหน่งเคอร์เซอร์ของเม้าส์ด้วยตัวแปร x,y ที่เก็บไว้ในบรรทัดที่ 14 ทำให้เม้าส์อยู่ที่เดิม

8. บรรทัดที่ 18 กำหนดให้ค่าที่รองรับการกดปุ่มเม้าส์ให้เป็น 0 (ถ้าค่าเป็น 1,2 จะมีการกดปุ่มของเม้าส์) และให้แสดงเคอร์เซอร์ของเม้าส์ ณ จุดนั้น ๆ

เมื่อคอมพายล์จะได้ยูนิต head.tpu เพื่อนำไปใช้งานต่อไป ซึ่งไฟล์ thai.tpu และ head.tpu นี้ ในตอนที่ 2-3 จะนำมาใช้ด้วย

```
 1: var  Lwindow,Rwindow,Uwindow,Dwindow:word;

 2: PROCEDURE WindowMouse(XLeft,Yup,Xright,Ydown:word);
 3: begin
 4:   regs.CX:=XLeft; regs.DX:=Xright;
 5:   regs.AX:=7;      Intr($33,regs);
 6:   regs.CX:=Yup;    regs.DX:=YDown;
 7:   regs.AX:=8;      Intr($33,regs);
 8:   Lwindow:=Xleft;  Rwindow:=Xright;
 9:   Uwindow:=Yup;    Dwindow:=YDown;
10: end;

11: PROCEDURE ClearMouse;
12: var x,y:integer;
13: begin
14:   FindXY(x,y);
15:   Regs.AX:=0; Intr($33,Regs);
16:   WindowMouse(Lwindow,Uwindow,Rwindow,Dwindow);
17:   setposition(x,y);
18:   Regs.BX:=0; mouseon;
19: end;
```

รูปที่ 1 ส่วนของชุดคำสั่งที่แก้ไขและเพิ่มเติมไฟล์ head.pas ของเดิม

สร้างยูนิตใช้งาน

เนื่องจากเราสร้างเครื่องคิดเลขเพื่อนำไปใช้ในการทำข้อสอบ เพื่อให้เกิดความสะดวกในการเรียกใช้ในภายหลัง จึงเขียนโปรแกรมในลักษณะยูนิตไฟล์ โดยผู้เขียนมีความตั้งใจในการสร้างเครื่องคิดเลขให้เป็นดังนี้

1. มีปุ่มกดแบบ 3 มิติจำนวน 6 แถว ๆ ละ 4 ปุ่ม

2. ปุ่มในการป้อนตัวเลขจะเหมือนกับเครื่องคิดเลขทั่ว ๆ ไปคือใช้เลข 0-9 และ .

3. ปุ่มสำหรับการคำนวณมี บวก ลบ คูณ หาร หาส่วนกลับของตัวเลขได้ ถอดรากที่สอง หาค่า sin, cos, tan และการรวมผลสุดท้าย (เครื่องหมาย =)

4. มีปุ่มเคลียร์ตัวเลข 2 แบบ คือ C สำหรับเคลียร์ตัวเลขที่ป้อนผิดเพียงบางหลัก และ AC สำหรับเคลียร์ตัวเลขทั้งหมดเพื่อเริ่มต้นการคำนวณชุดใหม่

5. การกดปุ่มของเครื่องคิดเลขนั้นจะเป็นการกดโดยอาศัยเม้าส์เข้ามาช่วย โดยไม่มีการกดด้วยแป้นพิมพ์แต่อย่างใด (คือผู้ใช้จะต้องติดตั้งเม้าส์ด้วย)

6. โปรแกรมจะแสดงการยุบตัวของแป้นที่กดให้เห็นบนจอภาพด้วย เพื่อให้เกิดความสมจริง

เรามาดูตัวแปรใช้งานและค่าคงที่ที่ใช้ในโปรแกรมก่อน ซึ่งคือรูปที่ 2 มีรายละเอียดดังนี้

1. บรรทัดที่ 1 กำหนดชื่อยูนิต ซึ่งต้องตรงกับชื่อของไฟล์ด้วย (ได้ยูนิตชื่อ calculat.tpu)

2. บรรทัดที่ 2-5 กำหนดส่วนที่เรียกใช้ได้จากภายนอกยูนิต

3. บรรทัดที่ 6-12 ค่าคงที่ 2 มิติเพื่อนำไปเขียนปุ่มต่าง ๆ ในเครื่องคิดเลขที่สร้าง ให้สังเกตว่ามีบางค่าได้กำหนดรหัสแอสกี 32 ไว้ เพราะว่าไม่สามารถกำหนดเป็นอักขระในตอนนี้ได้ โดยจะไปกำหนดค่าให้ใหม่ในตอนหลัง ค่าคงที่นี้จะนำไปแสดงบนปุ่มบนเครื่องคิดเลขอีกทีหนึ่ง

4. บรรทัดที่ 13 กำหนดตำแหน่งของปุ่มกดของเครื่องคิดเลขในแกน x

5. บรรทัดที่ 14 กำหนดตำแหน่งของปุ่มกดของเครื่องคิดเลขในแกน y

6. บรรทัดที่ 15-22 ตัวแปรใช้งานที่เกี่ยวข้อง โดยตัวแปรที่สำคัญ 2 ตัว แปรคือ Xcal และ Ycal เป็นตัวแปรที่ใช้ในการกำหนดให้เครื่องคิดเลขไปปรากฏที่ส่วนต่าง ๆ บนจอภาพได้ รวมทั้งจะใช้ในการตรวจสอบว่าผู้ใช้ได้กดปุ่มใดของเครื่องคิดเลข

7. บรรทัดที่ 23-27 ชุดคำสั่งวาดรูปสี่เหลี่ยมทึบ (สีหมายเลข 7) โดยกำหนดให้หยุดแสดงเคอร์เซอร์เม้าส์ก่อนวาดรูปสี่เหลี่ยม

8. บรรทัดที่ 28-29 ชุดคำสั่งเรียกใช้ไฟล์ข้อมูลอักขระภาษาอังกฤษจากไฟล์ litt.obj ซึ่งเป็นไฟล์ที่โอนข้อมูลมาจากไฟล์ litt.chr โดยอาศัยไฟล์ binobj.exe (ไฟล์ทั้งหมดมีในต้นฉบับของเทอร์โบปาสคาลทุกรุ่น)

```
 1: UNIT Calculat;
 2: Interface
 3: uses crt,dos,graph,thai,Head;
 4: PROCEDURE Calculator(XXCal,YYCal:word);
 5: Implementation

 6: const key:array[1..6,1..4] of char=
 7:         (('7','8','9','-'),
 8:          ('4','5','6','x'),
 9:          ('1','2','3','-'),
10:          ('0','.','=','+'),
11:          (' ',' ',' ',' '),
12:          (' ',' ','C',' '));
13:       Xkey:array[1..4] of word=(10,36,62,88);
14:       Ykey:array[1..6] of word=(45,70,95,120,145,170);
15: var Total,NumInput,Result:real;
16:     ST:string;
17:     ck:integer;    Mode:char;
18:     Xcal,Ycal:word;
19:     Start,Error,Special,ReadFile,SaveFile:boolean;
20:     dot,Xchoice,YChoice:byte;
21:     Pic1,pic2:array[1..1000] of byte;
22:     f:file;

23: PROCEDURE BBar(x1,y1,x2,y2:word);
24: begin
25:   Mouseoff; setfillstyle(1,7);
26:   Bar(x1,y1,x2,y2); Mouseon;
27: end;

28: PROCEDURE LittFont; External;
29:  {$L Litt.obj}
```

รูปที่ 2 ส่วนหัวและตัวแปรใช้งานในยูนิต Calculat.tpu

การเปลี่ยนไฟล์ litt.chr ให้เป็นไฟล์ litt.obj ทำได้ดังนี้

```
A:\ >binobj litt.chr litt littfont
```

ซึ่งเมื่อจบการทำงานจะได้ไฟล์ litt.obj ในแผ่นบันทึก โดยไฟล์นี้จะต้องกำหนดชุดคำสั่งเพื่อเรียกใช้งานในชื่อ littfont เท่านั้น (ชื่อตามที่เราตั้งเปลี่ยนเป็นชื่ออื่นก็ได้)

การตรวจสอบการกดแป้นเครื่องคิดเลข

ลักษณะของเครื่องคิดเลขที่ได้เขียนโปรแกรมนี้ ประกอบด้วยแป้นต่าง ๆ จำนวน 6 แถว ๆ ละ 4 แป้น ตามค่าคงที่ key (บรรทัดที่ 6-12) ผู้เขียนใช้ตัวแปร Xchoice เก็บคอลัมน์ และ Ychoice เก็บค่าแถว หากเราหาค่า Xchoice และ Ychoice ได้ เราก็ตรวจได้ว่าเป็นการกดแป้นใดของเครื่องคิดเลข

โดยไปเทียบกับค่าคงที่ key (ค่าของตัวแปรนี้บางค่าจะถูกกำหนดใหม่ในตอนรันโปรแกรม)

โปรแกรมนี้ทำงานด้วยเม้าส์เท่านั้น ทั้งนี้เพราะบนจอภาพจะมีปุ่มที่ใช้ในการคำนวณทางคณิตศาสตร์ เช่น COS, SIN, TAN, 1/X ด้วย ดังนั้นผู้อ่านที่ไม่มีเม้าส์คงต้องพัฒนาบางส่วนของโปรแกรมเอง

ชุดคำสั่งในการตรวจสอบการกดปุ่มใดของเครื่องคิดเลขคือโปรแกรมในรูปที่ 3 มีรายละเอียดดังนี้

1. บรรทัดที่ 33 หาตำแหน่งที่กดปุ่มของเม้าส์ด้วยตัวแปร Xmouse, Ymouse โดยค่าที่ได้นี้จะเป็นค่าจริงของจอภาพ (การ setviewport จะไม่ทำให้ตำแหน่งนี้ย้ายตามค่าใหม่ที่ตั้ง)

2. บรรทัดที่ 34 หาค่าตำแหน่งจริงของ Xmouse, Ymouse เมื่อเทียบกับเครื่องคิดเลขต้นแบบ ซึ่งจะกำหนดให้มุมบนซ้ายเป็น 0,0 ดังนั้นเมื่อเรากำหนดค่าตัวแปร Xcal, Ycal ที่ไม่ใช่ 0 จึงหาค่าจริงด้วยการคำนวณดังนี้ Xmouse จะเท่ากับ Xmouse-Xcal และ Ymouse-Ycal คือ Ymouse นำค่าทั้งสองนี้ไปตรวจการกดปุ่มเครื่องคิดเลขต่อไป

3. บรรทัดที่ 35-36 กำหนดค่าเริ่มต้นให้กับตัวแปร

4. บรรทัดที่ 37-43 วนรอบตรวจสอบหาค่า Xmouse ซึ่งถ้าค่า Xmouse อยู่ในขอบเขตของคอลัมน์ที่กำหนดด้วยค่าคงที่ Xkey (บรรทัดที่ 13 ของรูปที่ 2) ก็จะได้ค่า Xmouse และจบการวนรอบ หรือถ้าไม่อยู่ในขอบเขตเลยก็จบการวนรอบเมื่อ k = 5

5. บรรทัดที่ 44-49 คำนวณหาค่าของตัวแปร Ychoice ซึ่งเป็นการกำหนดขอบเขตตรวจสอบแต่ละเงื่อนไขโดยตรง

```
30: PROCEDURE CheckKey;
31: var k:byte;
32: begin
33:  FindXy(Xmouse,YMouse);
34:  XMouse:=Xmouse-Xcal;  YMouse:=Ymouse-Ycal;
35:  XChoice:=0; YChoice:=0;
36:  k:=1; ch:=#0;
37:  repeat
38:   if (Xmouse>Xkey[k]) and (Xmouse<Xkey[k]+20) then
39:     begin
40:      Xchoice:=k; k:=5;
41:     end;
42:    inc(k);
43:  until k>4;
44:  if (Ymouse>45) and (Ymouse<65) then Ychoice:=1;
45:  if (Ymouse>70) and (Ymouse<90) then Ychoice:=2;
46:  if (Ymouse>95) and (Ymouse<115) then Ychoice:=3;
47:  if (Ymouse>120) and (Ymouse<140) then Ychoice:=4;
48:  if (Ymouse>145) and (Ymouse<165) then Ychoice:=5;
49:  if (Ymouse>170) and (Ymouse<190) then Ychoice:=6;
50:  if (XChoice in [1..4]) and (Ychoice in [1..6]) then
51:    begin
52:      ch:=Key[Ychoice][XChoice];     MouseOff;
53:      getimage(Xkey[Xchoice],Ykey[YChoice],Xkey[Xchoice]+20,
54:                                          Ykey[YChoice]+20,pic1);
55:      getimage(Xkey[Xchoice]+3,Ykey[YChoice]+3,
56:                          Xkey[Xchoice]+17,Ykey[YChoice]+17,pic2);
57:      setfillstyle(1,6);
58:      Bar(Xkey[Xchoice],Ykey[YChoice],Xkey[Xchoice]+20,Ykey[YChoice]+20);
59:      putimage(Xkey[Xchoice]+3,Ykey[YChoice]+3,pic2,0); delay(120);
60:      putimage(Xkey[Xchoice],Ykey[YChoice],pic1,0);  MouseOn;
61:    end;
62: end;
```

รูปที่ 3 ชุดคำสั่งในการตรวจสอบว่ากดปุ่มของเม้าส์ที่แป้นใดของเครื่องคิดเลข

ผู้อ่านลองแก้ไขโปรแกรมส่วนนี้โดยอาศัยตัวแปร Ykey (บรรทัดที่ 14 รูปที่ 2)

6. บรรทัดที่ 50-61 ถ้าการกดปุ่มของเม้าส์อยู่ในขอบเขตของปุ่มใดปุ่มหนึ่งในเครื่องคิดเลข (อยู่ในช่องว่างระหว่างปุ่มของเครื่องคิดเลขจะได้ Xmouse, Ymouse เท่ากับ 0) จะทำงานด้วยคำสั่งกลุ่มนี้

7. บรรทัดที่ 52 ให้ค่า ch เท่ากับค่าคงที่ key ที่คำนวณได้ และสั่งให้เคอร์เซอร์เม้าส์หยุดแสดง (ซึ่งเป็นการสั่งให้คืนภาพเดิม ณ ตำแหน่งเคอร์เซอร์เม้าส์อยู่ให้กับจอภาพ)

8. บรรทัดที่ 53-54 จำภาพในขอบเขตของปุ่มที่กดในตัวแปร pic1

9. บรรทัดที่ 55-56 จำภาพขนาดเล็กกว่าขอบเขตในบรรทัดที่ 53-54 เล็กน้อยในตัวแปร pic2

10. บรรทัดที่ 57-58 วาดรูปสี่เหลี่ยมด้วยสีหมายเลข 6 ในขอบเขตเดียวกับภาพของตัวแปร pic1

11. บรรทัดที่ 59 ลงภาพ pic2 พร้อมกับหน่วงเวลา ส่วนนี้จะทำให้ผู้ใช้

โปรแกรมมองดูเหมือนว่าปุ่มได้ถูกกดให้ยุบลงไป

12. บรรทัดที่ 60 คืนภาพ pic1 ให้กับจอภาพ ทำให้ได้ปุ่มเดิมของเครื่องคิดเลขคืนมา พร้อมกับแสดงเคอร์เซอร์ของเม้าส์

การอ่านแฟ้มเอกสาร

แม้ว่าบทความชุดนี้ผู้เขียนประสงค์ที่จะให้เป็นโครงงานในเรื่องการทำข้อสอบแบบปรนัย แต่เพื่อให้ได้งานที่คุ้มค่ามากที่สุด ผู้เขียนจึงเขียนโปรแกรมให้เครื่องคิดเลขนี้สามารถนำไปใช้งานแบบเอกเทศด้วยตนเองได้ และยังสามารถนำไปใช้งานร่วมกับโปรแกรมพิมพ์เอกสารได้ด้วย (ผู้เขียนทดสอบกับราชวิถีเวิร์ดพีซี)

เพื่อให้ใช้งานร่วมกับโปรแกรมพิมพ์เอกสาร ผู้เขียนกำหนดให้อ่านแฟ้มเอกสารข้างนอกชื่อ text ได้ เพื่อที่จะนำตัวเลขและลักษณะการคำนวณในไฟล์นี้มาแสดงบนจอภาพได้

โปรแกรมในการอ่านแฟ้มเอกสารเป็นดังรูปที่ 4 มีรายละเอียดในชุดคำสั่ง readtext ดังนี้

```
63: PROCEDURE Readtext;
64: var Data:array[0..4000] of char;
65:     size,k:word; no:byte;
66: begin
67:  assign(f,'Text');
68:  {$I-} reset(f,1); {$I+}
69:  if IOResult=0 then
70:    begin
71:      setviewport(0,0,getmaxx,getmaxy,clipon);
72:      size:=filesize(f); if size>4000 then size:=4000;
73:      blockread(f,data,size);
74:      close(f);
75:      k:=0;  st:='';  No:=0;  setcolor(11);
76:      clsgr(127,0,getmaxx,getmaxy);
77:      repeat
78:       case Data[k] of
79:        #32..#228:ST:=ST+Data[k];
80:        #10:begin
81:              outthaixy(16,No*2+1,ST);
82:              ST:='';     Inc(no);
83:            end;
84:       end;
85:       inc(k);
86:      until (no>16) or (k>size);
87:      if ST<>'' then outthaixy(16,No*2+1,ST);
88:      setviewport(Xcal,Ycal,getmaxx,getmaxy,clipon);
89:      ReadFile:=false;
90:    end;
91: end;
```

รูปที่ 4 ส่วนการอ่านข้อความ ตัวเลขจากไฟล์ชื่อ text เข้ามาแสดงบนจอภาพ

1. บรรทัดที่ 64-65 กำหนดตัวแปร data เพื่อรองรับการอ่านข้อมูลจากแฟ้มเอกสาร text และตัวแปรใช้งาน

2. บรรทัดที่ 67 กำหนดชื่อไฟล์ที่ต้องการอ่าน

3. บรรทัดที่ 68 เปิดไฟล์ โดยจะมีการตรวจความผิดพลาดด้วย

4. บรรทัดที่ 69-90 ถ้าไฟล์นี้มีจริงจะทำงานด้วยคำสั่งกลุ่มนี้

5. บรรทัดที่ 71 กำหนดขอบเขตของจอภาพให้เต็มจอภาพ

6. บรรทัดที่ 72 ตรวจความยาวของไฟล์ และกำหนดขนาดของไฟล์ถ้าพบว่ามีความยาวเกิน

7. บรรทัดที่ 73-74 อ่านข้อมูลจากแฟ้มเอกสารและปิดแฟ้ม

8. บรรทัดที่ 75 กำหนดตัวแปรเริ่มต้น

9. บรรทัดที่ 76 ลบจอภาพทางด้านขวาของเครื่องคิดเลข

10. บรรทัดที่ 77-86 วนรอบแสดงข้อความที่อ่านได้บนจอภาพ โดยอาศัยตัวแปร ST เก็บข้อความและแสดงข้อความบนจอภาพเมื่อพบรหัสจบบรรทัด (รหัสแอสกี 10)

11. บรรทัดที่ 87 หากพบว่าข้อความที่ตรวจสอบยังไม่ได้แสดงบนจอภาพให้แสดงข้อความนั้น

12. บรรทัดที่ 88 กำหนดขอบเขตของจอภาพให้ตรงกับตำแหน่งของเครื่องคิดเลข

13. บรรทัดที่ 89 กำหนดให้อ่านแฟ้มนี้ได้เพียงครั้งเดียว

แฟ้มเอกสาร text ถ้ามีข้อความจะต้องเป็นรหัสเกษตรเท่านั้น การอ่านแฟ้มเอกสารนี้จะทำได้เมื่อให้เครื่องคิดเลขอยู่ทางด้านซ้ายของจอภาพเท่านั้น เหตุผลคือ ถ้าเครื่องคิดเลขนี้อยู่ในระบบการทำข้อสอบไม่ต้องการให้อ่านแฟ้มเอกสารนี้

การเรียกใช้ชุดคำสั่งนี้จะเกิดขึ้นเมื่อมีการกดแป้น F1

เก็บผลการคำนวณ

ผลการคำนวณในการใช้เครื่องคิดเลขนี้ ผู้เขียนคิดว่าควรถูกโอนไปยังแฟ้มเอกสารได้ เพื่อผู้ใช้โปรแกรมพิมพ์เอกสารจะนำไปแสดงบนแฟ้มเอกสารที่ประสงค์จะคำนวณ จึงได้กำหนดส่วนการเก็บผลการคำนวณไว้ในแฟ้มเอกสารชื่อ result ดังโปรแกรมในรูปที่ 5 มีรายละเอียดดังนี้

1. บรรทัดที่ 93 กำหนดตัวแปรใช้งาน

2. บรรทัดที่ 95 เปิดไฟล์ชื่อ result

3. บรรทัดที่ 96-106 ในกรณีที่เป็นครั้งแรกในการเปิดไฟล์จะทำงานตามคำสั่งส่วนนี้ โดยจะทำการเปลี่ยนค่าตัวแปร result ให้เป็นข้อความ STT และเขียนข้อความนี้ลงในแฟ้มเอกสาร result โดยมีรหัสจบบรรทัดและรหัสท้ายแฟ้มลงไปด้วย และปิดแฟ้มเอกสาร

4. บรรทัดที่ 108-117 ในกรณีที่เป็นการเขียนแฟ้มเอกสารครั้งต่อไปจะทำงานด้วยคำสั่งกลุ่มนี้ โดยจะตรวจสอบว่าแฟ้มนี้มีจริงในบรรทัดที่ 109 และเลื่อนตัวชี้ท้ายแฟ้มเอกสารขึ้นมาหนึ่งไบต์ พร้อมกับเขียนผลการคำนวณที่ได้ลงแฟ้มเอกสารและปิดแฟ้มนี้

ตัวเลขที่เก็บลงแฟ้มเอกสาร result นี้จะเป็นตัวเลขที่เห็นบนจอภาพของเครื่องคิดเลขในขณะนั้น ซึ่งการเรียกใช้ชุดคำสั่งนี้จะเกิดขึ้นเมื่อมีการกดแป้น F2

ในกรณีที่ผู้ใช้งานร่วมกับแฟ้มเอกสาร (สมมติใช้ราชวิถีเวิร์ดพีซี) จะทำงานได้ดังนี้

1. ขณะอยู่ในโปรแกรมพิมพ์แฟ้มเอกสาร เมื่อมีตัวเลขที่ต้องการคำนวณให้กด ^KB, ^KK เพื่อทำการกำหนดขอบเขตข้อความให้อยู่ในแถบสว่างก่อนตามด้วยกด ^KS และให้ชื่อแฟ้มเอกสารว่า text จะเป็นการเก็บข้อความที่กำหนดนี้ลงแผ่นบันทึกในไฟล์ชื่อ text

2. กดแป้น ^KO และป้อน minical เพื่อเรียกใช้ไฟล์ minical.exe

3. เมื่อไฟล์ minical.exe ทำงานให้กดแป้น F1

4. ทำการเก็บผลการคำนวณด้วยการกดแป้น F2

5. เมื่อการคำนวณสิ้นสุดให้กด OFF เพื่อเลิกการใช้เครื่องคิดเลข

6. โปรแกรมจะกลับมาที่โปรแกรมพิมพ์เอกสารอีกครั้ง

7. กด ^KR ป้อนชื่อแฟ้ม result จะได้ผลการคำนวณเข้ามาสู่แฟ้มเอกสาร

```
 92: PROCEDURE SaveResult;
 93: var STT:string; k:byte;
 94: begin
 95:   assign(f,'Result');
 96:   if pred(SaveFile) then
 97:     begin
 98:      {$I-} rewrite(f); {$I+}
 99:      if IOResult=0 then
100:       begin
101:        reset(f,1);  SaveFile:=true;
102:        str(Result:12:2,STT);  STT:=STT+#13#10#26;
103:        for k:=1 to length(STT) do  blockwrite(f,STT[k],1);
104:        close(f);
105:       end;
106:     end
107:      else
108:       begin
109:        {$I-} reset(f,1); {$I+}
110:        if IOResult=0 then
111:         begin
112:          seek(f,filesize(f)-1);
113:          str(Result:12:2,STT); STT:=STT+#13#10#26;
114:          for k:=1 to length(STT) do  blockwrite(f,STT[k],1);
115:         close(f);
116:        end;
117:       end;
118: end;
```

รูปที่ 5 ชุดคำสั่งเก็บผลการคำนวณลงในไฟล์ result

การคำนวณ

คราวนี้มาถึงการประมวลผลตัวเลขที่ได้จากการกดปุ่มต่าง ๆ ของเครื่องคิดเลข เนื่องจากการทำงานอยู่ในโหมดกราฟิก เพื่อความสะดวกเราจึงถือว่าการป้อนตัวเลขนั้นเป็นการป้อนข้อความก่อนเพราะสามารถแก้ไขการป้อนได้ง่ายกว่า โดยทุกครั้งที่เราป้อนข้อความก็จะเปลี่ยนค่าให้เป็นตัวเลขไว้ในตัวแปรที่กำหนดนำไปใช้ในการประมวลผลต่อไป

การประมวลผลในเครื่องคิดเลขของโปรแกรมนี้ไม่ใช่มีแค่การบวกลบคูณหารเท่านั้น ยังมีการหาค่าทางตรีโกณมิติและส่วนกลับของค่าตัวเลข

นั้น ๆ ด้วย ดังนั้นเราจึงแยกเขียนส่วนการประมวลผลนี้ออกมาเป็นชุดคำสั่งหนึ่งต่างหาก เพื่อให้การตรวจสอบของเราง่ายขึ้น

โปรแกรมในรูปที่ 6 คือชุดคำสั่งในการประมวลผลที่ได้กำหนดไว้ มีรายละเอียดดังนี้

1. บรรทัดที่ 119-131 ชุดคำสั่งสำหรับแสดงข้อความบนจอภาพ โดยจะเริ่มต้นด้วยการลบจอภาพของเครื่องคิดเลขก่อน ตามด้วยการเขียนข้อความจากการป้อน (ตัวแปร ST) ลงบนจอภาพ และจบลงด้วยการเปลี่ยนข้อความให้เป็นตัวเลขในตัวแปร NumInput

2. บรรทัดที่ 136-140 การทำงานเมื่อผู้ใช้กดที่ปุ่ม AC (ของเครื่องคิดเลข) ซึ่งจะเป็นการเคลียร์ตัวเลขทั้งหมดและเริ่มต้นการคำนวณตัวเลขกลุ่มใหม่

3. บรรทัดที่ 141-154 ส่วนการเคลียร์ตัวเลขที่กำลังป้อน โดยเป็นการตัดตัวเลขส่วนที่อยู่ขวามือออกไป โดยจะเลื่อนตัวเลขที่อยู่ทางซ้ายมือไปทางขวาหนึ่งหลักอย่างต่อเนื่อง

4. บรรทัดที่ 155-158 จบการทำงานของโปรแกรมทั้งหมดเมื่อกดที่ปุ่ม OFF (ของเครื่องคิดเลข)

5. บรรทัดที่ 159-162 การคำนวณเพื่อหาส่วนกลับของตัวเลข เมื่อกดปุ่ม 1/X (ของเครื่องคิดเลข)

6. บรรทัดที่ 163-171 คำนวณหาค่ารากที่สองของตัวเลขที่ปรากฏบนเครื่องคิดเลข ซึ่งผู้เขียนใช้แนวทางจากเครื่องคิดเลขที่ผู้เขียนมี โดยจะสามารถหาค่านี้ได้แม้ว่าตัวเลขจะเป็นจำนวนลบ

7. บรรทัดที่ 172-174 การคำนวณเมื่อต้องการหาค่า SIN

8. บรรทัดที่ 175-177 การคำนวณเมื่อต้องการหาค่า COS

รูปที่ 6 ชุดคำสั่งแสดงตัวเลขที่ป้อน และส่วนการคำนวณ

```
119: PROCEDURE shownum;
120: var k:byte; x:word;
121: begin
122:  Bbar(12,18,108,35);
123:  if ST[0]>#12 then
124:   begin
125:    delete(ST,1,length(ST)-12);  Error:=true;
126:    Bbar(20,10,108,20);  outtextxy(40,11,'ERROR');
127:   end;
128:  x:=20; if ST[0]=#12 then X:=12;  setcolor(0);
129:  for k:=1 to length(ST) do Outthaixy(x+(k-1)*8,15,#17+ST[k]);
130:  val(ST,NumInput,ck);
131: end;

132: PROCEDURE Re_sult(var No,Dotck:byte);
133: var kk:byte;
134: begin
135:   case ch of
136:    'A':begin
137:          ST:='          0';   showNum;  No:=0;
138:          Start:=True;   ToTal:=0;   Mode:=#32;
139:          Bbar(20,10,108,20);  Dot:=0;
140:        end;
141:   'C':begin
142:          if ST<>'          0' then
143:             begin
144:              if ST[11]='.' then Dotck:=0;
145:              for kk:=11  downto 2 do ST[kk]:=ST[kk-1];
146:              ST[1]:=' ';
147:              if ST='           ' then
148:               begin
149:                 ST:='          0'; no:=0;
150:               end;
151:              ShowNum;
152:             end
153:              else No:=0;
154:        end;
155:     'O':begin
156:           setviewport(0,0,getmaxx,getmaxy,clipon);
157:           closegraph; halt;
158:         end;
159:     'I':begin
160:           val(ST,NumInput,ck);
161:           if NumInput<>0 then NumInput:=1/Numinput;
162:         end;
163:     'R':begin
164:           val(ST,NumInput,ck);
165:           if NumInput>=0 then NumInput:=sqrt(Numinput)
166:            else
167:             begin
168:              NumInput:=sqrt(abs(Numinput));
169:              Numinput:=-NumInput;
170:             end;
171:         end;
172:     'S':begin
173:           val(ST,NumInput,ck);  NumInput:=SIN(Numinput/180*pi);
174:         end;
175:     'c':begin
176:           val(ST,NumInput,ck);  NumInput:=COS(Numinput/180*pi);
177:         end;
```

9. บรรทัดที่ 178-181 การคำนวณเมื่อต้องการหาค่า TAN

10. บรรทัดที่ 183-189 ในกรณีที่หาค่าส่วนกลับตัวเลข ค่ารากที่สอง ค่า SIN ค่า COS และ ค่า TAN จะทำงานตามคำสั่งกลุ่มนี้ด้วย โดยเป็นการตรวจสอบจำนวนเลขหลังจุดทศนิยมพร้อมกับให้แสดงผลด้วยชุดคำสั่ง shownum

การใช้เครื่องคิดเลข

เรื่องที่ต้องเขียนโปรแกรมต่อไป คือ ส่วนของการป้อนตัวเลขหรือส่วนการเลือกวิธีการประมวลผล โดยจะต้องแสดงเคอร์เซอร์ของเม้าส์ให้เห็นบนจอภาพในช่วงการรอกดปุ่มของเครื่องคิดเลข เมื่อมีการกดปุ่มของเม้าส์ที่ตำแหน่งใด ๆ ก็ตามบนส่วนของเครื่องคิดเลข จะต้องตรวจสอบได้ว่ากดที่ปุ่มใดด้วย

โปรแกรมในรูปที่ 7 จะเป็นชุดคำสั่งในเรื่องนี้ มีรายละเอียดของโปรแกรมดังนี้

1. บรรทัดที่ 194-195 กำหนดค่าให้ตัวแปรเพื่อเริ่มต้นการทำงาน

2. บรรทัดที่ 202 ทำการเคลียร์การกดปุ่มของเม้าส์

3. บรรทัดที่ 203-213 ทำการวนรอบจนกว่าจะมีการกดปุ่มของเม้าส์

4. บรรทัดที่ 209 ถ้ากดแป้น F2 เป็นการเก็บผลการคำนวณลงแฟ้มเอกสารชื่อ result

5. บรรทัดที่ 210 ถ้ากดแป้น F1 จะอ่านแฟ้มเอกสารชื่อ text มาแสดงบนจอภาพ

6. บรรทัดที่ 214 ตรวจสอบว่าเป็นการกดปุ่มใดของเครื่องคิดเลข

7. บรรทัดที่ 215-224 การ

รูปที่ 6 (ต่อ)

```
178:       'T':begin
179:             val(ST,NumInput,ck);
180:             NumInput:=SIN(Numinput/180*pi)/COS(Numinput/180*pi);
181:           end;
182:    end;
183:     if ch in ['I','R','S','c','T'] then
184:      begin
185:        Str(NumInput:11:2,ST);
186:        if (ST[length(ST)-1]='0') and
187:           (ST[length(ST)]='0') then Str(NumInput:11:0,ST);
188:        showNum;  Special:=true;
189:      end;
190: end;
```

รูปที่ 7 ส่วนการทำงานเมื่อกดปุ่มของเม้าส์ในการใช้เครื่องคิดเลข

```
191: PROCEDURE InputNum;
192: var kk,Dotck,k,ckk,no:byte;
193: begin
194:  NumInput:=0;  No:=0;   ST:='          0';
195:  ckk:=0;  DotCk:=0;  Special:=false;
196:  if Error then
197:   begin
198:     Bbar(20.10,108,20); outtextxy(40,11,'ERROR');
199:   end;
200:  repeat
201:   repeat
202:     clearMouse;
203:     repeat
204:       regs.AX:=3; Intr($33,regs);
205:      if Keypressed then
206:       begin
207:        readchar;
208:        case funckey of
209:         #60:SaveResult;
210:         #59:if ReadFile then Readtext;
211:        end;  {CASE Funckey}
212:       end; {Keypressed}
213:     until Regs.BX<>0;
214:      CheckKey;
215:      if Error then
216:       begin
217:        if Ch='=' then
218:         begin
219:          Bbar(12,10,108,35); Error:=false; Total:=0;
220:          NumInput:=0; ST:='          0';  ckk:=0;
221:          DotCk:=0;   showNum;  No:=0;
222:         end;
223:          ch:=#0;
224:       end;
225:      if (Dotck=1) and (ch='.') then ch:=#0;
226:      if (ST='          0') and (ch='0') then ch:=#0;
227:   until ch in ['0'..'9','.','-','+','*','=','/','A',
228:                                'C','I','O','R','S','c','T'];
229:    if Special and (ch in ['0'..'9']) then
230:      begin
```

รูปที่ 7 (ต่อ)

```
231:          Special:=false;    Start:=True; ckk:=0;
232:          ST:='          0';   showNum;  No:=0;
233:        end;
234:      if (Dotck=0) and (ch='.') then DotCk:=1;
235:      if (ch in ['I','R','S','c','T']) and (No=0) then
236:                                        str(Total:11:2,ST);
237:     case ch of
238:      '0'..'9','.':begin
239:                 if ST[1]=' ' then
240:                  begin
241:                   if (ckk=0) and start then
242:                    begin
243:                     Bbar(12.18.102,35); ckk:=27; Total:=0;
244:                    end;
245:                   if No=0 then
246:                    begin
247:                     No:=27; ST[11]:=' ';
248:                     if ch='.' then ST[11]:='0';
249:                    end;
250:                  for kk:=1 to 10 do ST[kk]:=ST[kk+1];
251:                  ST[11]:=ch;  ShowNum;
252:                 end;
253:               end;
254:      'A'..'c':Re_sult(No,Dotck);
255:     end;
256:      Result:=NumInput;
257:    until ch in ['-','+','*','=','/'];
258:     val(ST,NumInput,ck);
259: end;
```

ทำงานเมื่อเกิดความผิดพลาดในการป้อน ซึ่งในโปรแกรมจะเกิดจากผลการคำนวณแล้วได้ตัวเลขมากกว่า 12 หลัก

8. บรรทัดที่ 225 ตรวจให้การกดทำได้ครั้งเดียวต่อการป้อนตัวเลขหนึ่งจำนวน

9. บรรทัดที่ 226 ตรวจส่วนการกดปุ่ม 0 (ของเครื่องคิดเลข) ในกรณีที่ยังไม่ได้กดปุ่มตัวเลขอื่น

10. บรรทัดที่ 227 ส่วนของเงื่อนไขในการออกจากการวนรอบการป้อน

11. บรรทัดที่ 229-233 ในกรณีที่เป็นการคำนวณฟังก์ชันพิเศษ (SIN, COS ฯลฯ) ถ้ามีการกดแป้นตัวเลข 0-9 (ของเครื่องคิดเลข) จะเป็นการเริ่มต้นป้อนเลขจำนวนใหม่

12. บรรทัดที่ 234 ตรวจให้การกด . ทำได้ครั้งเดียวเมื่อ Dotck = 0

13. บรรทัดที่ 238-253 การทำงานเมื่อกดปุ่มตัวเลข 0-9 (ของเครื่องคิดเลข)

14. บรรทัดที่ 254 ทำการประมวลผลโดยเรียกชุดคำสั่ง Re__sult

15. บรรทัดที่ 256 กำหนดให้ตัวแปร 2 ตัวมีค่าเท่ากัน เพื่อที่จะใช้งานเมื่อกดแป้น F2

16. บรรทัดที่ 258 เปลี่ยนข้อความ ST ให้เป็นจำนวนตัวเลขเก็บไว้ในตัวแปร NumInput

แสดงผลการคำนวณ

เมื่อได้นำตัวเลข 2 จำนวนมาทำการประมวลผลแล้ว เราจะต้องตรวจสอบขอบเขตของตัวเลขที่ป้อนเพื่อให้สามารถแสดงบนจอภาพของเครื่องคิดเลขได้ ดังนั้นจึงต้องเขียนชุดคำสั่งเพื่อการทำงานในเรื่องนี้ โดยได้โปรแกรมดังรูปที่ 8 มีรายละเอียดดังนี้

1. บรรทัดที่ 264 กำหนดข้อความในตัวแปร STT ให้ว่าง พร้อมกับลบจอภาพของเครื่องคิดเลข

2. บรรทัดที่ 265 เปลี่ยนผลการคำนวณ (ตัวแปร total) ให้เป็นข้อความโดยมีตัวเลขหลังจุดทศนิยมตามค่าตัวแปร dot

3. บรรทัดที่ 266-280 ถ้าตัวแปร ST (เป็นค่าที่ป้อนหรือคำนวณได้) มากกว่า 12 หลัก จะหาตัวเลขไว้ในตัวแปร multi เพื่อลดตำแหน่งการแสดงผลลงและแสดงค่าตัวเลขที่มาคูณกับจำนวนตัวเลขที่แสดงบนจอภาพเพื่อได้ค่าโดยประมาณบนจอภาพ

4. บรรทัดที่ 281-282 ถ้าค่าที่ป้อนไม่เกิน 12 หลักให้แสดงค่านั้นบนจอภาพ

5. บรรทัดที่ 283-294 ในกรณีที่ค่าที่ป้อนเกิน 12 หลัก จะวนรอบรอกดปุ่มของเม้าส์ ซึ่งต้องกดที่ปุ่ม AC หรือ = เพียง 2 ปุ่มเท่านั้น พร้อมกับเริ่มต้นการป้อนใหม่

วาดรูปเครื่องคิดเลข

ก่อนที่เราจะใช้โปรแกรมนี้ สิ่งแรกสุดที่ต้องทำคือการวาดรูปเครื่องคิดเลขให้ปรากฏบนจอภาพ ซึ่งเราใช้ตัวแปร Xcal, Ycal, Xkey, Ykey และค่าคงที่ key ช่วยในการกำหนดตำแหน่งต่าง ๆ ของปุ่มบนเครื่องคิดเลข โดยเราจะแสดงปุ่มต่าง ๆ เป็นแป้นนูนให้ดูเหมือนจริง ส่วนที่เป็นจอภาพของเครื่องคิดเลขนั้นเราจะวาดเป็นรูปแป้นที่ยุบลง

ไปจากพื้น

ข้อความบนแป้นบางแป้นเราไม่สามารถกำหนดบนค่าคงที่ key ได้ เนื่องจากมีจำนวนมากกว่าหนึ่งไบต์ เราก็จะใช้วิธีเขียนข้อความลงไปแทน โดยข้อความที่มีหลายไบต์จะใช้ฟอนต์อักขระจากไฟล์ litt.obj อีกที

เขียนโปรแกรมวาดรูปเครื่องคิดเลขได้ดังรูปที่ 9 มีรายละเอียดดังนี้

1. บรรทัดที่ 299 กำหนดตำแหน่งการแสดงเครื่องคิดเลขตามตัวแปร Xcal, Ycal

2. บรรทัดที่ 300 กำหนดขอบเขตให้เม้าส์เคลื่อนที่ได้เฉพาะส่วนที่เป็นเครื่องคิดเลขเท่านั้น

3. บรรทัดที่ 301 กำหนดตำแหน่งเริ่มต้นการแสดงเคอร์เซอร์ของเม้าส์

4. บรรทัดที่ 302 วาดรูปกระดาน 3 มิติให้เป็นเครื่องคิดเลข และวาดรูปกระดาน 3 มิติแบบยุบลงเพื่อให้เป็นจอภาพของเครื่องคิดเลข

5. บรรทัดที่ 304-311 วาดปุ่มต่าง ๆ ของเครื่องคิดเลข พร้อมกับแสดงข้อความจากตัวแปร key

6. บรรทัดที่ 312 แสดงข้อความของปุ่ม

7. บรรทัดที่ 313-314 แสดงข้อความของปุ่ม AC และข้อความของปุ่ม 1/X

8. บรรทัดที่ 315 กำหนดให้ใช้ฟอนต์อักขระจากไฟล์ litt.obj จากชุดคำสั่ง littfont โดยให้มีขนาดโต 4 เท่า

9. บรรทัดที่ 316 แสดงข้อความของปุ่ม SIN

10. บรรทัดที่ 317 แสดงข้อความของปุ่ม COS

11. บรรทัดที่ 318 แสดงข้อความของปุ่ม TAN

```
260: PROCEDURE ShowTotal;
261: var X,k:integer;  STT:string;
262:     multi:real;
263: begin
264:  STT:='';       Bbar(12,18,108,35);
265:  setcolor(0);  str(Total:12:Dot,ST);
266:  if ST[0]>#12 then
267:    begin
268:     X:=ord(ST[0])-12;    ST:=copy(ST,1,12);
269:     Bbar(20,10,108,20); Multi:=1;
270:     for k:=1 to X do  Multi:=Multi*10;
271:     str(Multi:10:0,STT); k:=10;
272:     repeat
273:      if STT[k]=#32 then
274:        begin
275:          STT[k]:='x'; k:=0;
276:        end;
277:      dec(k);
278:     until k<=0;
279:     for k:=1 to length(ST) do Outthaixy(12+(k-1)*8,15,#17+STT[k]);
280:   end;
281:  if STT='' then
282:    for k:=1 to length(ST) do Outthaixy(12+(k-1)*8,15,#17+ST[k]);
283:  if STT<>'' then
284:   begin
285:    repeat
286:      clearMouse;
287:        repeat
288:         regs.AX:=3; Intr($33,regs);
289:        until Regs.BX<>0;
290:      CheckKey;
291:      if ch='A' then Ch:='=';
292:    until ch='=';
293:     Total:=0; showTotal;
294:   end;
295: end;
```

รูปที่ 8 ชุดคำสั่งแสดงผลที่ได้จากการประมวลผล

12. บรรทัดที่ 319 แสดงข้อความของปุ่ม OFF

13. บรรทัดที่ 320 กำหนดฟอนต์อักขระเป็นฟอนต์เดิม

14. บรรทัดที่ 321-327 แสดงข้อความให้ทราบถึงการใช้แป้น F1, F2

15. บรรทัดที่ 328-330 วาดรูปเครื่องหมายรากที่สองบนปุ่มการหารากที่สอง

16. บรรทัดที่ 331-333 กำหนดค่าให้กับค่าคงที่ key บางค่าใหม่ ซึ่งจะใช้สำหรับการประมวลผลในกรณี การหาร การหาส่วนกลับ การคูณ การเลิกงาน การเคลียร์ตัวเลขทั้งหมด การหาค่า SIN, COS, TAN และรากที่สองตามลำดับ

โปรแกรมหลัก

ขั้นตอนสุดท้ายจึงเป็นการเขียนโปรแกรมหลักเพื่อให้การวาดรูปเครื่องคิดเลข การตรวจการกดปุ่มของเครื่องคิดเลข และการประมวลผลตามแนวทางที่เราคิดไว้ โดยสามารถกำหนดให้เครื่องคิดเลขอยู่ที่ตำแหน่งที่ต้องการได้ ส่วนของโปรแกรมใช้งานจึงประกอบด้วย

1. การตรวจสอบขอบเขตของ

การแสดงเครื่องคิดเลข

2. ส่วนตรวจการป้อนตัวเลข

3. ส่วนแยกตัวเลข 2 จำนวนให้เป็นจำนวนตัวตั้งและจำนวนที่นำมากระทำ

4. ตรวจสอบชนิดของจอภาพของเครื่องคอมพิวเตอร์ เพื่อความถูกต้องของการทำงาน โดยต้องเป็นจอสีหรือจอวีจีเอโมโนโครมเท่านั้น

ส่วนของโปรแกรมหลักจึงเป็นดังรูปที่ 10 มีรายละเอียดดังนี้

1. บรรทัดที่ 339 กำหนดให้ตัวแปร Xcal, Ycal เท่ากับค่าที่เรียกใช้ชุดคำสั่ง calculator

2. บรรทัดที่ 340-342 กำหนดขอบเขตของตัวแปร Xcal, Ycal ถ้าค่า Xcal>5 จะไม่มีการอ่านแฟ้มเอกสาร text จากการกดแป้น F1

3. บรรทัดที่ 343 แสดงเครื่องคิดเลข ณ ตำแหน่งที่กำหนด

4. บรรทัดที่ 347 ป้อนตัวเลขด้วยชุดคำสั่ง InputNum

5. บรรทัดที่ 348-351 ถ้าเป็นการเริ่มต้นจะกำหนดค่าให้กับตัวแปร total และ start

6. บรรทัดที่ 352-364 ส่วนการประมวลผลเมื่อเลือกการลบ บวก คูณ และหาร ตามลำดับ

7. บรรทัดที่ 365-368 แสดงผลการคำนวณบนจอภาพของเครื่องคิดเลข

8. บรรทัดที่ 369-385 การทำงานเมื่อกดแป้น บวก ลบ คูณ หาร และจบการประมวลผล (กด=) ซึ่งส่วนนี้เป็นการทำงานในครั้งแรกของการจบการป้อนเลขจำนวนหนึ่ง (ยังไม่ใช้การประมวลผล) โดยกำหนดให้ตัวแปร mode มีค่าเท่ากับตัวแปร ch เพื่อนำไปประมวลผลในตอนต่อไป

9. บรรทัดที่ 390-393 ตรวจการ

```
296: PROCEDURE ShowCalculator;
297: var y,k:byte;
298: begin
299:  setviewport(Xcal,Ycal,getmaxx,getmaxy,clipon);
300:  WindowMouse(Xcal+10,Ycal+44,Xcal+107,Ycal+187);
301:  setPosition(Xcal+10,Ycal+44);
302:  board3D(0,0,120,197,3,0);     board3D(8,7,112,38,7,1);
303:  setcolor(0); Error:=false;
304:  for y:=1 to 6 do
305:   begin
306:     for k:=1 to 4 do
307:      begin
308:       Board3D(Xkey[k],Ykey[y],Xkey[k]+20,Ykey[y]+20,6,0);
309:       Outtextxy(Xkey[k]+7,Ykey[y]+7,key[y][k]);
310:      end;
311:   end;
312:   outtextxy(95,52,'.');     outtextxy(95,47,'.');
313:   outtextxy(39,176,'AC');   outtextxy(90,172,'1');
314:   outtextxy(94,176,'/');    outtextxy(99,180,'X');
315:   settextstyle(RegisterBgiFont(@Littfont),0,4);
316:   outtextxy(11,150,'SIN'); outtextxy(12,150,'SIN');
317:   outtextxy(37,150,'COS'); outtextxy(38,150,'COS');
318:   outtextxy(63,150,'TAN'); outtextxy(64,150,'TAN');
319:   outtextxy(13,174,'OFF'); outtextxy(12,174,'OFF');
320:   settextstyle(0,0,1);
321:   if ReadFile then
322:    begin
323:     setcolor(14);
324:     OutTextxy(5,202,'F1=>Load Text');
325:     OutTextxy(5,212,'F2=>Save Result');
326:     setcolor(0);
327:   end;
328:   line(95,158,92,154);      line(95,158,98,150);
329:   line(98,149,104,149);     line(96,158,93,154);
330:   line(96,158,99,150);      line(98,150,104,150);
331:   key[1][4]:='/';   key[6][4]:='I';  key[2][4]:='*';
332:   key[6][1]:='O';   key[6][2]:='A';  key[5][1]:='S';
333:   key[5][2]:='c';   key[5][3]:='T';  key[5][4]:='R';
334:   settextstyle(0,0,1);
335: end;
```

รูปที่ 9 ชุดคำสั่งวาดรูปเครื่องคิดเลขแสดงปุ่มกดแบบ 3 มิติ

รูปที่ 10 โปรแกรมหลักแสดงเครื่องคิดเลขและการทำงานทั้งหมด

```
336: PROCEDURE Calculator(XXCal,YYCal:word);
337: var k,y:word;
338: begin
339:    XCal:=XXCal;  YCal:=YYCal;
340:    if XCal>510 then Xcal:=510;
341:    if YCal>260 then Ycal:=260;
342:    if Xcal>5 then ReadFile:=false;
343:    ShowCalculator;
344:    Total:=0; Mode:=#32; Start:=True; dot:=0;
345:    setcolor(0);  Outthaixy(100,15,#17'0'); Mouseon;
346:  repeat
347:    inputNum;
348:    if Start then
349:      begin
350:        Total:=NumInput; Start:=false;
351:      end;
```

รูปที่ 10 (ต่อ)

```
352:    case Mode of
353:     '-':Total:=Total-NumInput;
354:     '+':Total:=Total+NumInput;
355:     '*':Total:=Total*NumInput;
356:     '/':begin
357:           if NumInput<>0 then Total:=Total/NumInput
358:            else
359:             begin
360:               Error:=true;
361:               Total:=0; Mode:=#32; Start:=true;
362:             end;
363:         end;
364:    end;
365:   Str(Total:12:2,ST);  Result:=Total;
366:   if (ST[length(ST)-1]='0') and (ST[length(ST)]='0') then
367:                                          dot:=0 else Dot:=2;
368:   if pred(Error) then  ShowTotal;
369:   if ch in ['-','+','*','=',,'/'] then
370:     begin
371:      if ch='=' then
372:        begin
373:          Mode:=#32; Start:=True; ch:=#32;
374:        end;
375:        Bbar(20,10,108,20);
376:        Mode:=ch; setcolor(0);
377:      if ch='*' then ch:='x';
378:      if ch<>'/' then outtextxy(101,11,ch)
379:       else
380:        begin
381:         outtextxy(101,11,'-');
382:         outtextxy(101,6,'.');
383:         outtextxy(101,11,'.');
384:        end;
385:     end;
386:  until ch=#255;
387: end;
388: {============ CALCULAT.PAS (CALCULAT.TPU) ==============}
389: begin
390:  if Monoscn then
391:    begin
392:      closegraph; halt;
393:    end;
394:  if Pred(mouse) then
395:    begin
396:      board3D(100,120,530,230,3,0); setcolor(0);
397:      outthaixy(120,140,#17'โปรแกรมนี้ทำงานด้วยเมาส์ เขียนโดยอาจหาญ สัตยารักษ์');
398:      outthaixy(120,180,#17'   โปรดรันไฟล์ Driver ของเมาส์ก่อนการรันไฟล์นี้   ');
399:      readchar;
400:      closegraph; halt;
401:    end;
402:  ReadFile:=true;  SaveFile:=false;  setgraphmode(2);
403: end.
```

ไม่ทำงานบนจอโมโนโครมแบบธรรมดา (จอเขียว) โดยออกจากกราฟิกและจบโปรแกรม

10. บรรทัดที่ 394-401 ตรวจการรันไดรเวอร์ของเม้าส์

11. บรรทัดที่ 402 กำหนดตัวแปรบูลีนที่ใช้งาน และกำหนดให้ใช้กราฟิกขนาด 640×480 จุด

เมื่อเก็บโปรแกรมทั้งหมด (ในรูปที่ 2-10) ลงในไฟล์ชื่อ calculat.pas แล้วทำการคอมพายล์ลงแผ่นบันทึกจะได้ยูนิตไฟล์ชื่อ calculat.tpu มาหนึ่งไฟล์นำไปใช้งานได้

```
1: PROGRAM Minical;
2: uses crt,dos,graph,thai,head,Calculat;
3: begin
4:  Calculator(0,50);
5: end.
```

รูปที่ 11 โปรแกรมสั้น ๆ เพื่อกำหนดให้เครื่องคิดเลขปรากฏ ณ ตำแหน่งที่กำหนด

โปรแกรมใช้งาน

ในยูนิตที่เขียนไปแล้วนี้ยังไม่สามารถรันเพื่อให้เป็นเครื่องคิดเลขได้ เราจึงต้องเขียนโปรแกรมสั้น ๆ ขึ้นมาอีกหนึ่งโปรแกรม โดยผู้เขียนให้ชื่อไฟล์ว่า minical.pas โดยจะต้องเรียกใช้ยูนิตไฟล์ที่เราได้พัฒนาขึ้นนี้ด้วยคือ head.tpu, thai.tpu และ calculat.tpu โดยในโปรแกรมจะเรียกใช้ชุดคำสั่ง calculator มาทำงานเพียงคำสั่งเดียวเท่านั้น

โปรแกรมจะเป็นดังรูปที่ 11 ซึ่งผู้เขียนคงไม่ต้องเขียนคำอธิบายก็คงเข้าใจได้ ถ้าต้องการให้เครื่องคิดเลขไปแสดงที่ตำแหน่งอื่นก็ให้เปลี่ยนตัวเลขได้ตามต้องการ

เมื่อคอมพายล์จะได้ไฟล์ minical.exe นำไปใช้งานได้ ผู้เขียนไม่ได้ลอกภาพของเครื่องคิดเลขมาให้ดู เพราะต้องการให้ผู้อ่านเห็นเองเมื่อรันโปรแกรมไฟล์ minical.exe นี้

ผู้อ่านคงต้องลองใช้เครื่องคิดเลขที่สร้างขึ้นนี้ไปพลาง ๆ ก่อน

พบกันใหม่ในฉบับต่อไปด้วยตอนที่ 2 สร้างแบบทดสอบ ซึ่งเป็นการป้อนข้อความประกอบกับกราฟิกอย่างง่าย ๆ คือ วาดรูปสี่เหลี่ยม วาดรูปวงกลมหรือวงรี และลากเส้นตรง เป็นต้น

สำหรับฉบับนี้ โชคดีครับ...